

Dear Sylvie, thank you to you and the steering group reviewers. We have addressed the concerns of the Steering Group review as written below in blue for your assessment.

Andy's comments

I did spot one or two minor issues as a standard reviewer, which are flagged in the attached copy.

Other things to be addressed:

- One other issue for the PEFf group to comment on is the apparent lack of a validator, which doesn't seem ideal for a PSI standard. I see there is a validator hosted on PeptideAtlas but I don't know if this has been updated, or has a formal status with regards to the standard. It isn't referenced from the spec doc (or if it is I can't find the reference)

The validator at PeptideAtlas is indeed up to date. This validator has now been mentioned in the specification document in the abstract and section 3.5.4, mentioned more prominently at the main PEFf web page, and the source code has been made available.

*- The PSI webpage for PEFf could do with some updates, and a hard-coded URL:
<http://www.psidev.info/node/363>*

The official URL for PEFf is <http://psidev.info/peff>. This has been made clearer in the specification document, including directly in the abstract. Furthermore, the web page has been updated so that the status is current.

Martin's comments

In addition to Andys comments (also added as comments to the .doc, there are some more minor changes tracked):

- The DocProc also wants the working group being stated, "if applicable". I think, here it is the Mass Spectrometry WG.

The Mass Spectrometry Standards Working Groups is now explicitly stated at the top.

- DocProc: "Data format standards should, where applicable be accompanied by a reference implementation". For other standards there was a programmatic API, but from my point of view it could be the validator or Comet, or maybe nextProt has made or can make their exporter open source?

One of the benefits of the PEFf format is that it is quite similar to the FASTA format. Since most relevant tools already have a FASTA parser, it seems likely that many software packages will simply update their FASTA parser to extract whatever additional information they wish, rather than using an external package. Nonetheless, we have referred to the Perl validator as the reference implementation, since it is likely the most strict parser. The new section 3.5.5 describes the reference implementation.

- In other formats we had a mapping table between MIAPE fields and XML elements (<http://www.psidev.info/mzidentml-conformance-miape>). Is that also possible here with FASTA elements? Maybe instead of a table just one to half a dozen sentences are enough.

The only relevant MIAPE requirements (specifically MIAPE-MSI requirements) regarding the database (aside from the fact that the database itself must be provided/identified) are the specification of a description, the version of the database, and the number of entries. All these concepts are supported in PEFf via the CV terms DbName, DbDescription, DbVersion, and NumberOfEntries. We have added this statement to the specification.

- Although actually an action point for mzIdentML: Have you already checked, whether referencing a PEFf file as database is encodable there? Or have you already added a CV term? If Yes, that could be a comment here. (same for mzTab).

Yes, it is possible to refer to a PEFf file in mzIdentML in the same way that a FASTA file is specified. See GitHub issue #6 for further discussion here: <https://github.com/HUPO-PSI/PEFF/issues/6>